REMARKS

Claims 1-32 are pending in the present application, with Claims 1, 14, and 25 as being the independent claims. In summary of the outstanding Office Action, claims 1-32 stand rejected.

Claim Rejections - 35 USC § 102

Claims 1-32 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Lee et al. (US 2002/0169788). Applicants note that Lee has a priority date of Feb. 16, 2000. Again without conceding whether Lee is prior art, Applicants nevertheless submit that it fails to anticipate the claimed invention.

As pointed out in the previous response, claim 1 recited "creating a record in a first buffer associated with the first relational table; and copying the record from the first buffer to the first relational table." This claim clearly requires creating a record in a buffer. The buffer is associated with a relational table, but *is not* the relational table itself. Thereafter, the *record* is copied from the *buffer* to a *relational table.*

The examiner in response to this argument, which was made in the previous response, indicates that:

> Applicant argues Lee makes no mention of nodes or rows or columns of a table and reveals no mention of record creation or buffers or copying from buffers to tables.

Action p. 6. The examiner counters that record creation or copying from buffers to tables *is* taught by Lee and cites to paragraphs 96 and 110 of Lee. Moreover, the examiner maintains that "Lee teaches a loader which loads the XML data contained in the document into the tables of a relational database." Action at 7. The applicants don't disagree that Lee teach loading XML data into the tables of a relational database. Nevertheless, that doesn't address the Applicants point, namely, the paragraphs of Lee cited by the examiner say NOTHING regarding copying records from a buffer to a table.

Applicants provide paragraphs 96 and 110 of Lee herein below to further illustrate the lack of such teaching:

> [0096] It is an important feature of this invention that the DTD 18 (i.e., the second document definition portion 18) is loaded by the system 10 and used in metadata format to generate the relational schema of the second data definition portion 22. Then, the XML data stored in the first data portion 16 of the XML document 12 is loaded by the system 10 into the tables making up the first data storage portion 20.
>
> ....
>
> [0110] Step 44 of loading the XML data 16 of the document 12 into the tables 20 of the relational database 14 according to the relational schema 22 generated in step 42 preferably comprises the step of loading the XML data 16 contained in the document 12 into the tables 20 of the relational database 14 according to the relational schema 22 generated herein (shown by reference number 60 and described in greater detail in FIGS. 12-13).

The specification describes this aspect of the invention and distinguishes from other schemes that load XML into relational tables as follows:

> As XML document 302 is shred, records for various tables are sorted into buffers associated with each table, e.g., buffer BL1 506 is associated with table 39, buffer BL2 508 is associated with table 37, and buffer BL3 510 is associated with table 35. Switch 502 determines which buffers, e.g., 506, 508, 510, get which records, and also controls when the records are written from various ones of the buffers, e.g., 506, 508, 510, into the associated tables, e.g., 39, 37, 35, respectively.
>
> Bulk Load accomplishes the shredding process "in situ", that is, it must interpret the hierarchical data, e.g., XML data, determine the destination SQL target fields and tables, and pass the resultant records to the server – all as it is encountering the XML data in the input stream. This is contrast to other XML to SQL insertion mechanisms such as Updategrams, which can load the entire sql:before and sql:after images of the data into memory, run an analysis on it to determine the affected records, then issue a sequence of SQL statements to effect the change. In order to work similar to Updategrams, Bulk Load would have to load the XML file and create the in-memory DOM for the data set. This is expensive for data sets involving thousands, or perhaps even millions, of records.

Specification, p. 13.

Accordingly, Applicants again respectfully disagree that claim 1 is anticipated by Lee and submit that the examiner failed to show a *prima facie* case of anticipation under 35 U.S.C. § 102(e).

Similarly, Applicants submit that claim 14 recites in part:

> mapping the hierarchical data based on the schema and creating records from the hierarchical data from nodes associated identified as data to be stored in the at least one column in each of the at least two relational tables; and streaming the records into the at least two relational tables.

The above limitations are not found in Lee. And claim 25 recites in part:

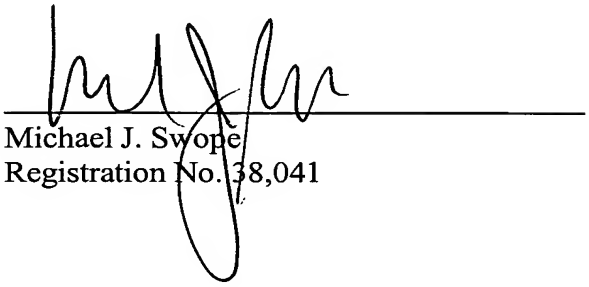
> instructions for mapping the hierarchical data based on the schema and creating records from the hierarchical data from nodes associated identified as data to be stored in the at least one column in each of the at least two relational tables; and instructions for streaming the records into the at least two relational tables.

Applicants submit that Lee fails to teach all of the limitations of the claimed invention. For at least the above reasons, the applicants submit that the reference falls short of anticipation and request that the examiner reconsider the rejection of the claims in view of Lee.

CONCLUSION

For the foregoing reasons, Applicants respectfully request reconsideration and allowance of claims 1-32 and issuance of a Notice of Allowability.

Date: June 30, 2005



Michael J. Swope
Registration No. 38,041

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439